



Content provided in partnership with [Prentice Hall PTR](#) from the book [PHP 5 Power Programming](#) by [Andi Gutmans](#), [Stig Bakken](#), and [Derick Rethans](#)

## Introduction

Following the huge success of its predecessors, PHP 5 brings many significant improvements to the most popular web scripting language on the globe. The book [PHP 5 Power Programming](#), written by Stig Bakken, Derick Rethans, and myself, provides readers with an in-depth view of this new PHP version right from its source.

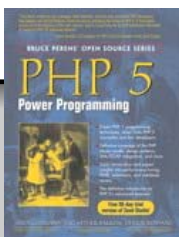
The foundation of PHP 5 is the new Zend Engine II scripting engine, which not only features a completely new and feature rich object oriented paradigm, but also provides the infrastructure for other significant changes such as great improvements in PHP's support for XML technologies. This article gives you a very short overview of PHP 5 and what makes it so great.

## New OO Features

The OO model of PHP 5 has been completely rewritten. Not only does the move from objects being native types to objects being handles give much more clarity on when objects are created, cloned, and destroyed, but it also allowed us to implement many new OO features which are common in traditional OO languages. There are far too many features to cover, but the following list gives you the highlights:

- As mentioned, PHP objects are now based on handles, which allows for clear-cut object constructors, clone methods and destructors
- The PHP object syntax can be completely overloaded by PHP extension authors, allowing the efficient implementation of important new features such as SimpleXML and the native SOAP extension, both of which are described later on
- Both properties and methods support the common OO public/private/protected access modifiers (as well as final and static)
- Abstract methods, abstract classes and interfaces
- Exception handling ala try/throw/catch
- Method return value de-referencing such as `$obj->method1()->method2()`

This list is only the tip of the iceberg, and there are many more OO improvements, too numerous to cover in this short overview.



Buy This Book From informIT

## ***XML Rewrite***

The support of XML technologies has been almost entirely rewritten. Besides having changed all the XML extensions to use the excellent Gnome libxml2 library and significant improvements in supporting standards such as better DOM compliance, the two most exciting additions in PHP 5 are SimpleXML, an extension which allows the manipulation of XML as native PHP objects, and the new native SOAP extension, which gives enterprise-grade access to web services.

## ***SimpleXML***

As mentioned, SimpleXML enables you to manipulate XML data as if it were a native PHP object. This mostly prevents you from having to deal with XML parsing standards such as SAX and DOM, improving XML productivity and lowering the barrier of entry into the XML manipulation world. Consider the following example:

```
people.xml :
<people>
<person sex="male">Joe</person>
<person sex="female">Judy</
person>
</people>
print_people.php:
<?php
$people = simplexml_load_file(
"people.xml");
foreach ($people->person as $
person) {
print $person . " is " . $per
son["sex"] . "\n";
}
?>
```

The result of this code is:

```
Joe is male
Judy is female
```

I assume you are convinced that there never has been an easier way to access and manipulate XML data. Not only can the SimpleXML extension be used to read XML data, but it can also be used to create and change XML data. If there is some special XML manipulation you need to do in DOM, you can do most of the basic work in SimpleXML and then convert it to DOM and run additional manipulations without having to write and re-read the XML data to some kind of temporary or permanent storage. This is due to the SimpleXML and DOM extensions working with the same libxml2 library, and special functions exist which convert SimpleXML representation to DOM and vice-versa.

## SOAP

PHP 5 features new native support of the SOAP protocol allowing PHP applications to easily access and expose web services on the Internet. Taking advantage of the Zend Engine II capabilities for overloading the OO syntax of PHP, accessing SOAP has never been easier.

The following code snippet shows how easy it is to call a web service method using its WSDL description file:

```
<?php
$client = SoapClient("some.wsdl");
$result = $client->someFunction();
?>
```

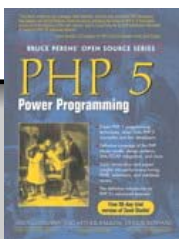
## Conclusion

It'd be easy to go on and on about PHP 5's new functionality. This article only covers a small amount of the new features but surely has given you a taste of PHP 5's new capabilities. You can find more information in the PHP 5 section of [Zend.com](http://Zend.com).

## About the Author

Andi Gutmans has been working on PHP since 1997, when along with Zeev Suraski he created PHP 3, and later on PHP 4. Recently, Andi has worked on the Zend Engine 2's object oriented improvements for the upcoming PHP 5 release. Andi is a member of both the PHP Group and the Apache Software Foundation. Additionally, he has a rich background in enterprise software development including real-time avionics simulation software and n-tier J2EE applications.

You can learn more about PHP in [PHP 5 Power Programming](#), by Andi Gutmans, Stig Bakken, and Derick Rethans, Prentice Hall PTR, ISBN 0-13-147149-X, ©2005



Buy This Book From informIT